第 1 章 IDA 数据显示窗口

本章内容

了解控制器的重要性

编写基本视图控制器

利用表视图控制器

你已经能够自信地❶将❷二❸进❹制❺文❻件❼加❽载❾到 IDA 中，一边喝着自己喜欢的饮料，一边让 IDA 发挥它的“魔力”。IDA 完成初始分析后，该是你接管控制权的时候了。熟悉 IDA 显示的最佳方法是，浏览 IDA 用于显示二进制数据的各种带标签的子窗口。对 IDA 越熟悉，执行逆向工程任务的效率也越高。

在详细介绍 IDA 的主要子窗口之前，首先了解 IDA 用户界面的如下基本规则会有所帮助。

如果由于你不小心按下某个键，导致数据库文件发生意外，这时，你必须自己将显示窗口恢复到以前的状态。

记住，IDA 的工具栏高度可配置，就像热键对菜单操作的映射一样 IDA 的工具栏高度可配置，就像热键对菜单操作的映射一样。

虽然这些菜单无法提供在某个位置允许执行的操作的详尽列表，但你可以用它们执行一些最常见的操作。

了解这些规则之后，下面开始介绍 IDA 主要的数据显示窗口。

5.1　IDA 主要的数据显示窗口

在默认配置下，IDA（从 6.1 版开始）会在对新二进制文件的初始加载和分析阶段创建 7 个显示窗口。这些窗口全部可以通过导航带下方显示的一组标题标签访问（如图 4-9 所示）。3 个立即可见的窗口分别为 IDA-View 窗口、函数窗口和消息输出窗口。无论这些窗口是否默认打开，我们在本章讨论的所有窗口都可通过 View ▸ Open Subviews 菜单打开。请记住这一点，因为你可能会经常无意中关闭 IDA 的显示窗口。

在 IDA 中，ESC 键是一个非常有用的热键。在反汇编窗口中，ESC 键的作用与 Web 浏览器的“后退”按钮类似，因此，它在导航反汇编窗口时非常有用（导航将在第 6 章详细介绍）。遗憾的是，在打开的其他窗口中，ESC 键用于关闭窗口。有时候，你可能恰恰想要关闭窗口，但其他情况下，你可能希望立即重新打开刚刚关闭的窗口。

5.1.1　反汇编窗口

1. IDA 图形视图

图 5-1 显示了图形视图中一个非常简单的函数。图形视图会让人联想到程序流程图，因为它将一个函数分解成许多基本块[①]，以生动显示该函数由一个块到另一个块的控制流程。

① 基本块是一个不包含分支，从头执行到尾的最大指令序列。因此，每个基本块都有唯一的入口点（块中的第一条

图片（Alt+T）

图 5-1 IDA 图形视图

在屏幕上你会发现，IDA 使用不同的彩色箭头区分函数块之间各种类型的流[①]。根据测试条件，在条件跳转位置终止的基本块可能会生成两种流：Yes 边的箭头（是的，执行分支）默认为绿色，No 边的箭头（不，不执行分支）默认为红色。只有一个后继块的基本块会利用一个正常边（默认为蓝色）指向下一个即将执行的块。

代码清单 5-1 控制表的创建方式和运行的 3 个方法

```
- (NSString *)tableView:(UITableView *)tableView
      titleForHeaderInSection:(NSInteger)section {
    if (section == 0) {
        return @"SDK Colors";
    } else if (section == 1) {
        return @"RGB Colors";
    }
    return 0;
}
```

设置单元格的文本和文本颜色

根据测试条件

在条件跳转位置终止的基本块可能会生成两种流：

注意 来源于希腊语是微小因此 uP 一般作为微处理器的缩写）来源于希腊语意思是微小（因此 uP 一般作为微处理器的缩写）。

指令）和退出点（块中的最后一条指令）。基本块中的第一条指令通常是分支指令的目标，而最后一条指令则往往是一条分支指令。

① IDA 使用术语流来表示某个指令如何继续执行。正常流（也叫做普通流）表示指令默认连续执行。跳转流表示当前的指令跳转到（或可能跳转到）某个非连续性位置。调用流表示当前指令会调用一个子例程。